Обобщение оценки расслоения-связности на случай произвольного графа Хассе

Фрей Александр, Решетняк Илья.

Содержание

1 Введение

[ToDo] Добавить введение.

Альтернативное название статьи: учёт верхней связности на основе слабого замыкания множества алгоритмов.

2 Основные обозначения

Пусть задана генеральная выборка $\mathbb{X} = (x_1, \dots, x_L)$, состоящая из L объектов. Произвольный алгоритм классификации, примененный к данной выборке, порождает бинарный вектор ошибок $a \equiv \bigl(I(a, x_i)\bigr)_{i=1}^{L}$, где $I(a, x_i) \in \{0, 1\}$ — индикатор ошибки алгоритма a на объекте x_i. В дальнейшем алгоритмы будут отождествляться с векторами их ошибок на выборке $\mathbb{X}$.

Обозначим через $\mathbb{A} = \{0, 1\}^L$ множество всех возможных векторов ошибок длины L. Через $[\mathbb{X}]^\ell$ обозначим множество всех разбиений генеральной выборки $\mathbb{X}$ на обучающую выборку X длины ℓ и контрольную выборку $\bar{X}$ длины $k = L - \ell$. *Число ошибок* алгоритма a на выборке $U \subseteq \mathbb{X}$ обозначим через $n(a, U) = \sum\limits_{x \in U} I(a, x)$. Величину $\nu(a, U) = n(a, U)/|U|$ будем называть *частотой ошибок* алгоритма a на выборке U. *Уклонение частот* на разбиении $\mathbb{X} = X \sqcup \bar{X}$ определим как разность частот ошибок на контроле и на обучении: $\delta(a, X) = \nu(a, \bar{X}) - \nu(a, X)$.

Пусть $A \subset \mathbb{A}$ — множество алгоритмов с попарно различными векторами ошибок. Обозначим через $A(X)$ множество алгоритмов с минимальным числом ошибок на обучающей выборке X:

$$A(X) = \operatorname*{Argmin}_{a \in A} n(a, X). \tag{1}$$

Частоту ошибок на обучающей выборке называют *эмпирическим риском. Минимизация эмпирического риска* μ — это метод обучения, который из заданного множества $A \subset \mathbb{A}$ выбирает алгоритм $a \in A$, допускающий наименьшее число ошибок на обучающей выборке X. Таким образом, для всех $X \in [\mathbb{X}]^\ell$ выполнено $\mu X \in A(X)$. В дальнейшем будет рассматриваться *пессимистическая* минимизация эмпирического риска, удовлетворяющая дополнительному условию $\mu X \in A(\mathbb{X})$ — то есть среди алгоритмов в $A(X)$ выбирается алгоритм с наибольшим числом ошибок на полной выборке.

Говорят, что метод μ переобучен на разбиении $X \sqcup \bar{X}$, если уклонение частот $\delta(a, X)$ превышает фиксированный порог ε. Переобучение может быть следствием «неудачного» разбиения генеральной выборки на обучение и контроль. Поэтому вводится функционал *вероятности переобучения*, равный доле разбиений выборки, при которых возникает переобучение [?, ?]:

$$Q_\varepsilon(A) = \mathsf{E}[\delta(\mu X, X) \geqslant \varepsilon], \text{ где } \mathsf{E} = \frac{1}{C_L^\ell} \sum_{X \in [\mathbb{X}]^\ell}.$$

Тут и далее квадратные скобки — нотация Айверсона, переводящая логическое выражение в число 0 или 1 по правилам [истина] = 1, [ложь] = 1.

Функционал $Q_\varepsilon(A)$ уже не зависит от выбора разбиения и характеризует качество данного метода обучения на данной генеральной выборке.

3 Теорема о порождающих и запрещающих объектах

Первый подход, позволивший получать точные оценки вероятности переобучения в рамках слабой вероятностной аксиоматики, основан на выделении порождающих и запрещающих объектов [?].

Гипотеза 1. Пусть множество A, выборка $\mathbb{X}$ и детерминированный метод обучения μ таковы, что для каждого алгоритма $a \in A$ можно указать конечное множество индексов V_a, и для каждого индекса $v \in V_a$ можно указать порождающее множество $X_{av} \subset \mathbb{X}$, запрещающее множество $X'_{av} \subset \mathbb{X}$ и коэффициент $c_{av} \in \mathbb{R}$, удовлетворяющие условиям

$$\bigl[\mu X{=}a\bigr] = \sum_{v \in V_a} c_{av} \bigl[X_{av} \subseteq X\bigr] \bigl[X'_{av} \subseteq \bar{X}\bigr], \quad \forall X \in [\mathbb{X}]^\ell. \tag{2}$$

Введем для каждого алгоритма $a \in A$ и каждого индекса $v \in V_a$ обозначения:

$$\begin{aligned}
L_{av} &= L - |X_{av}| - |X'_{av}|;\\
\ell_{av} &= \ell - |X_{av}|;\\
m_{av} &= n(a, \mathbb{X}) - n(a, X_{av}) - n(a, X'_{av});\\
s_{av}(\varepsilon) &= \frac{\ell}{L}\,(n(a, \mathbb{X}) - \varepsilon k) - n(a, X_{av}).
\end{aligned}$$

В условиях гипотезы 1 справедливы следующие утверждения о вероятностях получения алгоритмов и вероятности переобучения:

Теорема 1. *Если гипотеза 1 справедлива, то для всех $a \in A$ вероятность получить в результате обучения алгоритм a равна*

$$P_a = \mathsf{P}[\mu X{=}a] = \sum_{v \in V_a} c_{av} P_{av};$$

$$P_{av} = \mathsf{P}[X_{av} \subset \bar{X}][X'_{av} \subset \bar{X}] = \frac{C_{L_{av}}^{\ell_{av}}}{C_L^\ell},$$

а вероятность переобучения $Q_\varepsilon(A)$ выражается по формуле

$$Q_\varepsilon(A) = \sum_{a \in A} \sum_{v \in V_a} c_{av} P_{av} H_{L_{av}}^{\ell_{av}, m_{av}}(s_{av}(\varepsilon)), \tag{3}$$

где $H_L^{\ell,m}(z) = \sum\limits_{s=0}^{\lfloor z \rfloor} \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell}$ — гипергеометрическая функция распределения.

Отметим, что в ряде случаев удается подобрать лишь такие множества порождающих и запрещающих объектов, для которых (2) выполнено лишь в виде неравенства:

$$\bigl[\mu X{=}a\bigr] \leqslant \sum_{v \in V_a} c_{av} \bigl[X_{av} \subseteq X\bigr] \bigl[X'_{av} \subseteq \bar{X}\bigr], \quad \forall X \in [\mathbb{X}]^\ell. \tag{4}$$

Очевидно, что в данном случае выражения (3) будет давать верхнюю оценку для вероятности переобучения $Q_\varepsilon(A)$. В следующем параграфе данный факт будет использован для вывода верхней оценки вероятности переобучения.

4 Оценка расслоения-связности для связных семейств алгоритмов

Для пары алгоритмов $a, b \in A$ обозначим через $\rho(a, b)$ хэммингово расстояние между векторами ошибок алгоритмов a и b. Введем на A естественное отношение порядка: $a \leqslant b$ тогда и только тогда, когда $I(a, x) \leqslant I(b, x)$ для всех $x \in \mathbb{X}$. Определим $a < b$ есть $a \leqslant b$ и $a \neq b$. Если $a < b$ и при этом $\rho(a, b) = 1$, то будем говорить, что a *предшествует* b, и записывать $a \prec b$.

Определение 1. *Графом расслоения-связности* множества алгоритмов A будем называть направленный граф (A, E) с множеством ребер $E = \{(a, b) \colon a \prec b\}$.

Каждому ребру $a \prec b$ графа расслоения-связности соответствует один и только один объект $x_{ab} \in \mathbb{X}$, такой что $I(a, x_{ab}) = 0$ и $I(b, x_{ab}) = 1$.

Определение 2 (Верхняя связность)**.** Обозначим через $X_u(a) = \{x_{ab} \in \mathbb{X} \colon a \prec b\}$ множество объектов, соответствующих ребрам графа расслоения-связности, *исходящим* из вершины a. *Верхней связностью* $u(a) = |X_u(a)|$ назовем мощность множества $X_u(a)$.

Определение 3 (Нижняя связность)**.** Обозначим через $X_d(a) = \{x_{ba} \in \mathbb{X} \colon b \prec a\}$ множество объектов, соответствующих ребрам графа расслоения-связности, *входящим* в вершину a. *Нижней связностью* $d(a) = |X_d(a)|$ назовем мощность множества $X_d(a)$.

Связность $u(a)$ (или $d(a)$) есть число способов изменить алгоритм a так, чтобы он стал делать на одну ошибку больше (или меньше). Связность можно интерпретировать как число степеней свободы семейства A в локальной окрестности алгоритма $a \in A$.

Определение 4 (Неполноценность алгоритма)**.** Обозначим через $X_q(a) = \{x_{cb} \colon c \prec b < a\}$ множество объектов, соответствующих всевозможным ребрам (c, b) на путях, ведущих к вершине a. *Неполноценностью* $q(a) = |X_q(a)|$ алгоритма $a \in A$ будем называть мощность множества $X_q(a)$.

Легко доказать, что если метод μ является пессимистической минимизацией эмпирического риска, то $X_a = X_u(a)$ и $\bar{X}_a = X_q(a)$ можно использовать в качестве порождающего и запрещающего множества:

$$\bigl[\mu X{=}a\bigr] \leqslant \bigl[X_u(a) \subseteq X\bigr]\bigl[X_q(a) \subseteq \bar{X}\bigr], \quad \forall X \in [\mathbb{X}]^\ell. \tag{5}$$

Следовательно, имеет место следующая верхняя оценка:

Теорема 2 (оценка расслоения-связности)**.** *Для произвольной выборки $\mathbb{X}$, пессимистического метода минимизации эмпирического риска μ и произвольного $\varepsilon \in (0, 1)$*

$$Q_\varepsilon(A) = \sum_{a \in A} \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell} H_{L-u-q}^{m-q,\ell-u}\bigl(\tfrac{\ell}{L}(m - \varepsilon k)\bigr), \tag{6}$$

где $u \equiv u(a)$ — верхняя связность, $q \equiv q(a)$ — неполноценность, $m = n(a, \mathbb{X})$ — число ошибок алгоритма a на генеральном множестве объектов.

Благодаря комбинаторному сомножителю $\frac{C_{L-u-q}^{\ell-u}}{C_L^\ell}$ вклад каждого алгоритма a в оценку Q_ε экспоненциально убывает с ростом неполноценности q и связности u.

5 Разреженная монотонная сеть

В данном параграфе рассматривается *разреженная монотонная сеть* — модельное семейство алгоритмов, демонстрирующее один из недостатков полученной выше оценки (6).

Мы начнём с плотной (не-разреженной) многомерной сети алгоритмов. Данное семейство является моделью параметрического *связного семейства алгоритмов*, предполагающего, что при непрерывном удалении каждой компоненты вектора параметров от оптимального значения число ошибок на полной выборке только увеличивается. Известно, что рассмотренная выше оценка расслоения-связности (6) дает точное значение вероятности переобучения для многомерной сети алгоритмов. Тем не менее, для разреженной монотонной сети, полученной удалением некоторой части алгоритмов из плотной монотонной сети, оценка расслоения-связности вырождается.

Введём целочисленный вектор индексов $\boldsymbol{d} = (d_1, \ldots, d_h) \in \mathbb{Z}^h$. Обозначим $\|\boldsymbol{d}\| = \max\limits_{j=1,\ldots,h} |d_j|$, $|\boldsymbol{d}| = |d_1| + \ldots + |d_h|$. На множестве векторов индексов введём покомпонентное отношение сравнения: $\boldsymbol{d} < \boldsymbol{d}'$, если $d_j \leqslant d'_j$, $j = 1, \ldots, h$, и хотя бы одно из неравенств строгое.

Определение 5. Множество алгоритмов $A = \bigl\{a_{\boldsymbol{d}}\bigr\}$, где $\boldsymbol{d} \geqslant 0$ и $\|\boldsymbol{d}\| \leqslant D$ называется *монотонной h-мерной сеткой алгоритмов длины D*, если существует $h \in \mathbb{N}$ и упорядоченные наборы объектов $X_j = \{x_j^1, \ldots, x_j^D\} \subset \mathbb{X}$, для всех $j = 1, \ldots, h$, а так же множества $U_1 \subset \mathbb{X}$ и $U_0 \subset \mathbb{X}$, такие что:

1) набор $\bigl\{U_0, U_1, \{X_j\}_{j=1}^h\bigr\}$ является разбиением множества $\mathbb{X}$ на непересекающиеся подмножества;

2) $a_d(x_j^i) = [i \leqslant d_j]$, где $x_j^i \in X_j$;

3) $a_d(x_0) = 0$ при всех $x_0 \in U_0$;

4) $a_d(x_1) = 1$ при всех $x_1 \in U_1$.

Обозначим $|U_1| = m$. Из определения следует, что $n(a_{\boldsymbol{d}}, \mathbb{X}) = m + |d|$. Алгоритм $a_{\boldsymbol{0}}$ является *лучшим в сетке*. Множество алгоритмов с равным числом ошибок $t + m = n(a_{\boldsymbol{d}}, \mathbb{X})$ называются *t-слоем* сетки.

Пример 1. Монотонная двумерная сетка при $m = 0$ и $L = 4$:

$$\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{c} \begin{array}{ccccccccc} a_{0,0} & a_{1,0} & a_{2,0} & a_{0,1} & a_{1,1} & a_{2,1} & a_{0,2} & a_{1,2} & a_{2,2} \end{array} \\ \left(\begin{array}{ccccccccc} 0 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} \\ \hline 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{array}\right) \end{array}$$

Определение 6. Пусть $\kappa \in \mathbb{N}$ — целочисленный параметр; $A = \{a_{\boldsymbol{d}}\}$ — h-мерная монотонная сетка длины κD; $m \equiv n(a_0, \mathbb{X})$. *Разреженной h-мерной монотонной сеткой $\ddot{A}$ разреженности κ и длины D* будем называть подмножество A, заданное условием:

$$\ddot{A} = \left\{a_{\boldsymbol{d}} \in A \,\middle|\, \boldsymbol{d} \in (\kappa\mathbb{Z})^h\right\}.$$

Отметим, что при $\kappa > 1$ граф смежности разреженной монотонной сетки состоит из изолированных точек. Следовательно, для всех алгоритмов семейства $u(a) = q(a) = 0$, и оценка расслоения-связности вырождается.

[ToDo] отразить картинку по вертикали и переделать её векторно

Пример 2. На рисунке 2 выделено подмножество двумерной монотонной сетки с параметром $D = 8$, соответствующее разреженной монотонной сетке с параметрами $\kappa = 2$, $D = 4$.

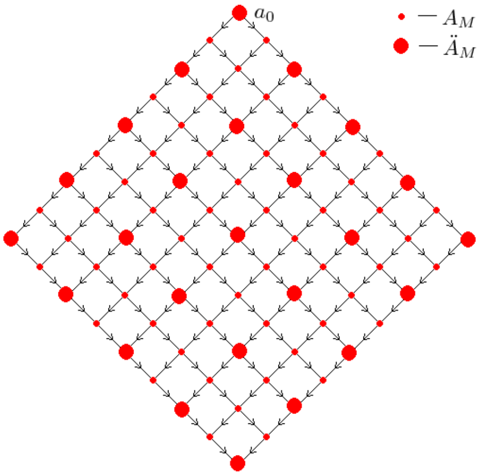

Рис. 1: Двумерная разреженная монотонная сетка при $\kappa = 2$, $D = 4$.

Тем не менее, для разреженной монотонной сети по-прежнему можно ввести систему порождающих и запрещающих множеств. Для этого нам потребуется следующее естественное обобщение графа расслоения-связности:

Определение 7. *Диаграммой Хассе* множества алгоритмов A называется ориентированный граф транзитивной редукции отношения $<$ частичного порядка на алгоритмах.

Легко проверить, для разреженной монотонной сети $\ddot{A}$ ребра диаграммы Хасса $E = \{(a,b)\colon a < b \text{ и } \rho(a,b) = \kappa\}$ соединяют те и только те пары алгоритмов, что находятся друг от друга на хэмминговом расстоянии κ, и идут в сторону увеличения числа ошибок алгоритмов на полной выборке.

В отличии от графа расслоения-связности, каждому ребру (a,b) диаграммы Хасса соответствует не один, а множество алгоритмов $X_{ab} = \{x \in \mathbb{X}\colon I(a,x) = 0 \text{ и } I(b,x) = 1\}$. Это позволяет обобщить понятия неполноценности в терминах диаграммы Хасса.

Определение 8 (Неполноценность алгоритма)**.** Обозначим через $X_q(a) = \bigcup_{b \leqslant a} X_{ba}$ множество объектов, соответствующих всевозможным ребрам диаграммы Хассе на всевозможных путях, ведущих к вершине a. *Неполноценностью* $q(a) = |X_q(a)|$ алгоритма $a \in A$ будем называть мощность множества $X_q(a)$.

Обозначим через $I_a = \{b \in A\colon (a,b) \in E\}$ множество концов ребер, исходящих из a. через $I^a = \{b \in A\colon (a,b) \in E\}$ — множество алгоритмов, из которых в a ведет ребро. Число исходящих из a ребер обозначим через $u(a) = |I_a|$.

Лемма 3. *Пусть A — произвольное множество алгоритмов, а метод обучения μ является пессимистической минимизацией эмпирического риска. Тогда необходимое условие получения произвольного алгоритма $a \in A$ в результате обучения записывается в терминах графа Хасса следующим образом:*

$$\big[\mu X{=}a\big] \leqslant \big[X_u(a) \subseteq \bar{X}\big] \prod_{b \in I_a} [X_{ab} \cap X \neq \emptyset], \quad \forall X \in [\mathbb{X}]^\ell. \tag{7}$$

Доказательство. TBD ■

Это условие означает следующее: для того, что бы алгоритм a был выбран методом обучения, необходимо и достаточно, что бы множество $X_u(a)$ целиком содержалось в контроле, а для каждого $b \in I_a$ хотя бы один объект из X_{ab} попал в обучение. В частном случае разреженной монотонной сети (8) обращается в равенство.

Обратим внимание, что в случае разреженной монотонной сети множества $\{X_{ab}\colon b \in I_a\}$, соответствующие различным b, попарно не пересекаются. Это важное свойство позволяет записать условие (8) на языке порождающих и запрещающих объектов.

Зафиксируем алгоритм $a \in A$, и пронумеруем элементы множества I_a произвольным способом: $I_a = \{b_1, \dots, b_{u(a)}\}$. Для каждого $b_i \in I_a$ пронумеруем элементы множества $X_{ab_i} = \{x_{i1}, \dots, x_{i\kappa}\}$ (тоже произвольным способом). Возьмем $V_a = \prod\limits_{b \in I_a} X_{ab}$ в качестве индексного множества, фигурирующего в гипотезе о порождающих и запрещающих объектах. Положим все $c_{av} = 1$. Элементы $v \in V_a$ будем записывать в виде вектора чисел: $v = (v_1, \dots, v_{u(a)})$, где все $v_i = 1, \dots, |X_{b_i}|$. Определим систему порождающих и запрещающих множеств следующим образом:

$$\bar{X}_{av} = \{x_{ij}\colon i = 1, \dots, u(a), j = 1, \dots, (v_i - 1)\} \cup X_q(a),$$
$$X_{av} = \{x_{ij}\colon i = 1, \dots, u(a), j = v_i\}.$$

Лемма 4. *Пусть метод μ является пессимистической минимизацией эмпирического риска, а множество алгоритмов A таково, что для каждого $a \in A$ множества $\{X_{ab}\colon b \in I_a\}$, соответствующие различным b, попарно не пересекаются. Тогда определенная выше система порождающих и запрещающих множеств даёт необходимое условие получение алгоритмов $a \in A$ в результате обучения:*

$$\big[\mu X{=}a\big] \leqslant \sum_{v \in V_a} \big[X_{av} \subseteq X\big] \big[X'_{av} \subseteq \bar{X}\big], \quad \forall X \in [\mathbb{X}]^\ell.$$

Доказательство. TBD ■

Следует отметить, что в лемме 4 условие о попарно-непересекающихся множествах X_{ab} является абсолютно искусственным техническим приемом. В следующем параграфе мы рассматриваем процедуру, достраивающую произвольное семейство A до $A^* \supset A$ так, что к A^* уже можно применять лемму 4.

6 Слабое замыкание семейства алгоритмов

Для алгоритма a обозначим через $\mathbb{X}_{[a]} \subset \mathbb{X}$ множество ошибок алгоритма a на выборке $\mathbb{X}$. Рассмотрим произвольное множество алгоритмов A и его диаграмму Хассе (A, E) — граф транзитивной редукции отношения $<$, определенного на парах алгоритмов условием «$a \leqslant b$ если $\mathbb{X}_{[a]} \subset \mathbb{X}_{[b]}$». Напомним также, что через $I_a = \{b \in A\colon (a, b) \in E\}$ обозначалось множество концов ребер, исходящих из a, через $u(a) = |I_a|$ — количество таких ребер, через $X_{ab} = \mathbb{X}_{[b]} \backslash \mathbb{X}_{[a]}$, где $a < b$ — множество объектов, соответствующих ребру (a, b). По аналогии с I_a и I^a обозначим через I_a^∞ множество алгоритмов b, таких что существует путь из a в b, проходящий по рёбрам графа Хассе; через I_∞^a — множество алгоритмов b, таких что существует путь из b в a. Для любой пары алгоритмов $a, b \in A$ определим алгоритм $a \cap b$ условием $\mathbb{X}_{[a \cap b]} = \mathbb{X}_{[a]} \cap \mathbb{X}_{[b]}$.

Будем говорить, что множество алгоритмов A является *слабо замкнутым*, если для любого $a \in A$ множества $\{X_{ab}\colon b \in I_a\}$ попарно не пересекаются. Наша задача — дополнить произвольное множество A алгоритмами до A^*, такого что $A \subset A^*$, и A^* — слабо замкнуто. Очевидно, что такое множество существует — достаточно взять $A^* = \{0, 1\}^L$. Следует отметить, что в данном случае неверно говорить о наименьшем по включению множестве. Действительно, легко построить пример двух слабо замкнутых множеств A_1, A_2, пересечение которых $A_1 \cap A_2$ уже не является слабо замкнутым. Именно этим неприятным свойством и объясняется термин *слабая* замкнутость.

Множество алгоритмов A назовём *замкнутым*, если для любого $a \in A$, и для любой пары $b_1, b_2 \in A$, такой что $a < b_1, a < b_2$, выполнено $(b_1 \cap b_2) \in A$. Легко показать, для любой пары замкнутых множеств их пересечение вновь является замкнутым. Это позволяет определить замыкание множества алгоритмов $\bar{A}$, как наименьшее по включению замкнутое множество, содержащее A.

Утверждается, что из замкнутости следует слабая замкнутость. В дальнейшем для произвольного множества A мы будем определять его слабое замыкание $A^* \subset \bar{A}$ с помощью описанной ниже алгоритмической процедуры.

Следующая лемма показывает, что вероятность переобучения для слабого замыкания семейства $Q_\varepsilon(A^*)$ всегда не меньше $Q_\varepsilon(A)$ и поэтому любую верхнюю оценку для $Q_\varepsilon(A^*)$ можно использовать как верхнюю оценку для $Q_\varepsilon(A)$.

Лемма 5. *Пусть A — произвольное множество алгоритмов, b - некоторый алгоритма, не принадлежащий A, но такой, что $\exists a \in A : a \leqslant b$, а метод обучения μ является пессимистической минимизацией эмпирического риска. Тогда $Q_\varepsilon(A \cup b) \geqslant Q_\varepsilon(A)$*

Доказательство. Рассмотрим множество разбиений $T(A)$ на которых ПМЭР переобучается. При добавлении в семейство нового алгоритма b на части из этих разбиений он может быть выбран методом обучения. На остальных разбиениях из $T(A)$ будут выбраны те же алгоритмы, и эти разбиения останутся в $T(A \cup b)$.

Рассмотрим произвольное разбиение $(X, \bar{X}) \in T(A)$, такое что при минимизации эмпирического риска для множества A был выбран алгоритм $c = \mu A$, и он оказался переобучен, а при минимизации эмпирического риска для множества $A \cup b$ был выбран алгоритм b.

Algorithm 1 Слабое замыкание множества алгоритмов

Вход: множество алгоритмов A, множество ребер графа Хассе E;

Выход: слабое замыкание A^*, новое множество ребер графа Хассе E^*;

1: Сгенерировать очередь заданий на обработку:
$Q := \{(a, b_1, b_2) \colon (a, b_1) \in E, (a, b_2) \in E\}$;
2: **пока** $Q \neq \emptyset$
3: $\quad (a, b_1, b_2) :=$ взять следующее задание из очереди Q;
4: $\quad$ **если** $(a, b_1) \notin E$ или $(a, b_2) \notin E$ **то перейти к шагу** 2;
5: $\quad c := b_1 \cap b_2$;
6: $\quad$ **если** $c \in A$ **то перейти к шагу** 2;
7: $\quad A := A \cup \{c\}$;
8: $\quad$ Найти множества $I_\infty^a, I_\infty^{b_1}, I_\infty^{b_2}, I_a^\infty, I_{b_1}^\infty, I_{b_2}^\infty$ для текущей пары (A, E).
9: $\quad I^c :=$ Т-Редукция $((I_\infty^{b_1} \cap I_\infty^{b_2}) \backslash I_\infty^a, <, \{a\})$;
10: $\quad I_c :=$ Т-Редукция $(I_a^\infty \backslash (I_{b_1}^\infty \cup I_{b_2}^\infty), >, \{b_1, b_2\})$;
11: $\quad E := E \cup \{(x, c) \colon x \in I^c)\} \cup \{(c, y) \colon y \in I_c)\}$;
12: $\quad$ **для всех** $x \in I^c$, $y \in I_c$
13: $\qquad$ **если** $(x, y) \in E$ **то**
14: $\qquad\quad E := E \backslash (x, y)$;
15: $\quad$ **для всех** $\{x, y\} \subset I_c$
16: $\qquad Q = Q \cup \{(c, x, y)\}$;
17: $\quad$ **для всех** $x \in I^c$
18: $\qquad$ **для всех** $y \in I_x$
19: $\qquad\quad$ **если** $y \neq c$ **то**
20: $\qquad\qquad Q = Q \cup \{(x, c, y)\}$;
21: Положить $A^* := A$, $E^* := E$.

Algorithm 2 Т-Редукция(Q, ϕ, R)

Вход: очередь кандидатов Q, предикат ϕ на парах из Q, начальное приближение R;

Выход: множество $\bar{R} \subset R \cup Q$, транзитивно-замкнутое относительно ϕ;

1: **пока** $Q \neq \emptyset$
2: $\quad x :=$ взять следующее задание из очереди Q;
3: $\quad D := \emptyset$;
4: $\quad$ **для всех** $y \in R$
5: $\qquad$ **если** $\phi(y, x)$ **то перейти к шагу** 1;
6: $\qquad$ **если** $\phi(x, y)$ **то** $D := D \cup \{y\}$;
7: $\quad R := (R \backslash D) \cup \{x\}$;
8: **вернуть** R

Для доказательства леммы нам достаточно показать, что на данном разбиении $(X, \bar{X})$ b переобучается. Для числа ошибок алгоритмов a, b, c на обучающей выборке X верны следующие соотношения:

$n(c, X) \leqslant n(a, X)$ (иначе из семейства A был бы выбран a).

$n(a, X) \leqslant n(b, X)$ $(a \leqslant b)$.

Следовательно $n(c, X) = n(b, X)$ (иначе из$A \cup b$ был бы выбран c).

Так как мы рассматриваем пессимистическую минимизацию эмпирического риска, то $n(b) >= n(c)$, если b был выбран вместо c.

Из неравенств $n(c, X) = n(b, X)$, $n(b) \geqslant n(c)$ следует что b переобучается, если переобучается c.

Лемма доказана. ■

В алгоритме построения слабого замыкания (6) на каждом шаге в семейство добавляется алгоритм b с непустым $X_q(b)$. Поэтому, согласно доказанной выше лемме,при построении слабого замыкания семейства, на каждом шаге вероятность переобучения только увеличивается, следовательно $Q_\varepsilon(A^*) \geqslant Q_\varepsilon(A)$.

7 Общая оценка расслоения-связности для разреженных семейств алгоритмов

Пусть множество алгоритмов A является слабо замкнутым.

Рассмотрим алгоритм $a \in A$ и систему множеств $\{X_{ab} \colon b \in I_a\}$, состоящую из наборов объектов, соответствующих выходящих из a ребрам диаграммы Хасса. Пронумеруем элементы множества I_a произвольным способом: $I_a = \{b_1, \dots, b_{u(a)}\}$, и рассмотрим вектор $w_a = (|X_{ab_i}|)_{i=1}^{u(a)}$. Напомним, что под модулем вектора $|w_a|$ мы понимаем сумму его координат. Рассмотрим также вектор $1_a = (1, \dots, 1)$, той же размерности что и w_a, но заполненный единицам. Рассмотрим функцию $S_a(u)$, определенную для всех u из $|1_a|, \dots, |w_a|$ выражением $S_a(u) = |\{w \in \mathbb{Z}^{u(a)} \colon |w| = u, 1_a \leqslant w \leqslant w_a\}|$. Значение $S_a(u)$ соответствует количеству векторов с целочисленными координатами, ограниченных снизу вектором 1_a, сверху - w_a, и с суммой координат u.

Теорема 6. *Пусть A — произвольное множество алгоритмов, A^* — его слабое замыкание. Тогда справедлива следующая оценка вероятности переобучения $Q_\varepsilon(A^*)$ и среднего значения числа ошибок на контроле $C(A^*)$:*

$$Q_\varepsilon(A^*) \leqslant \sum_{a \in A} \sum_{u=|1_a|}^{|w_a|} S_a(u) \frac{C_{L_a-u}^{\ell_a}}{C_L^\ell} H_{L_a-u}^{\ell_a, m_a}(s_a(\varepsilon)),$$

$$C(A^*) \leqslant \sum_{a \in A} \sum_{u=|1_a|}^{|w_a|} S_a(u) \frac{C_{L_a-u}^{\ell_a}}{C_L^\ell} \left(n(a, \mathbb{X}) - \frac{\ell_a}{L_a - u} m_a \right),$$

где введены следующие обозначения: $L_a = L - q(a)$, $\ell_a = \ell - u(a)$, $m_a = n(a, \mathbb{X}) - q(a)$, $s_a(\varepsilon) = \frac{\ell}{L}(n(a, \mathbb{X}) - \varepsilon k)$, $q(a)$ — неполноценность алгоритма a, определенная в терминах диаграммы Хассе.

Доказательство. Напомним, что согласно лемме 4 множества порождающих и запрещающих объектов можно выбрать следующим способом:

$$\bar{X}_{av} = \{x_{ij} \colon i = 1, \dots, u(a), j = 1, \dots, (v_i - 1)\} \cup X_q(a),$$
$$X_{av} = \{x_{ij} \colon i = 1, \dots, u(a), j = v_i\}.$$

Следовательно, мощности множеств $\bar{X}_{av}$ и X_{av} выражаются следующим способом: $|\bar{X}_{av}| = |v| - u(a) + q(a), |X_{av}| = u(a)$. Тогда, в обозначениях теоремы 1, получим:

$$\begin{aligned} L_{av} &= L - |v| - q(a) = L_a - |v|, \text{ где } L_a \equiv L - q(a), \\ \ell_{av} &= \ell - u(a) \equiv \ell_a, \\ m_{av} &= n(a, \mathbb{X}) - q(a) \equiv m_a, \\ s_{av} &= \frac{\ell}{L}(n(a, \mathbb{X}) - \varepsilon k) - 0 \equiv s_a, \end{aligned}$$

где выражения L_a, ℓ_a, m_a, s_a не зависят от v. Это позволяет записать сумму вида $\sum\limits_{v \in V_a} f(|v|)$ по декартовому произведению $V_a = \prod\limits_{b \in I_a} X_{ab}$ следующим образом:

$$\sum_{v \in V_a} f(|v|) = \sum_{u=|u(a)|}^{|w_a|} s_a(u) f(u),$$

где $s_a(u)$ — определенный выше коэффициент. ■

8 Оценка для случая пересекающихся рёбер

В данном параграфе мы рассмотрим альтернативный подход к учёту рёбер диаграммы Хассе в оценке расслоения-связности - прямое вычисление оценки для случая пересекающихся рёбер. Обозначим через $X_{u(a)} = \bigcup_{b \in I_a} X_{ab}$ - множество объектов, принадлежащих исходящим из a рёбрам.

Перепишем условие (8) в виде, необходимом для применения теоремы 1. Пусть $R(a) = \{r | r \subset X_u(a), \forall b \in I_a, X_{ab} \cap r \neq \emptyset\}$

$$\left[\mu X = a\right] \leqslant \sum_{r \in R(a)} \left[X_q(a) \cup (R(a) \backslash r) \subseteq \bar{X}\right] [r \subseteq X] \tag{8}$$

Тогда в обозначениях теоремы 1

$$\begin{aligned} L_{ar} &= L - |X_u(a)| - q(a); \\ \ell_{ar} &= \ell - |r|; \\ m_{ar} &= n(a, \mathbb{X}) - q(a); \\ s_{ar}(\varepsilon) &= s_a \\ P_{ar} &= \frac{C_{L_{ar}}^{\ell_{ar}}}{C_L^\ell} \end{aligned}$$

Запишем выражение для вклада одного алгоритма в оценку переобучения:

$$\sum_{r \in R(a)} P_{ar} H_{L_{ar}}^{\ell_{ar}, m_{ar}}(s_{ar}(\varepsilon)) \tag{9}$$

Заметим, что параметры $L_{ar}, \ell_{ar}, m_{ar}, s_{ar}, P_{ar}$ зависят только от мощности множества r, поэтому выражение можно переписать, сгруппировав слагаемые с одинаковой мощностью $|r|$:

$$\sum_{v=0}^{|X_u(a)|} T(v) \frac{C_{L-|X_u(a)|-q(a)}^{\ell-v}}{C_L^\ell} H_{L-|X_u(a)|-q(a)}^{\ell-v, n(a,\mathbb{X})-q(a)}(s_a(\varepsilon)) \tag{10}$$

,где $T(v) = \#\{r|r \in R(a), |r| = v\}$.

Задача, вычисления $T(v)$, вообще говоря, NP-трудна(она является обобщением задачи о покрытии множества), но эксперименты показывают, что число исходящих рёбер неединичной мощности у алгоритма обычно невелико, поэтому для вычисления $T(v)$ можно применять практически любой алгоритм. Мы предлагаем следующий алгоритм:

Пусть $x_1, \dots, x_{|X_u(a)|}$ - пронумерованные произвольным образом объекты из $X_u(a)$. Пусть $T(v, n, \alpha)$ - число подмножеств мощности v из объектов $x_1, \dots, x_n$, покрывающих множество рёбер $\alpha \subset I(a)$. Тогда для $T(v, n, \alpha)$ справедлива рекуррентная формула:

$$T(v, n, \alpha \cup \beta_n) = T(v, n-1, \alpha \cup \beta_n) + T(v-1, n-1, \alpha), \tag{11}$$

где $\beta_n = \{b|b \in I_a, x_n \in X_{ab}\}$.

Используя эту рекуррентную формулу можно рассчитать $T(v) \equiv T(v, |X_u(a)|, I(a))$ для всех значений v

9 Численный эксперимент

В данном параграфе рассматривается вопрос практической применимости полученных выше оценок вероятности переобучения. Оценка теоремы (6) применима лишь к слабо замкнутому множеству алгоритмов, поэтому в первую очередь требуется сравнить вероятности переобучения исходного множества алгоритмов A и его слабого замыкания A^*, построенного алгоритмом 6.

Отметим, что слабая замкнутость множества алгоритмов необходима лишь для учёта верхней связности. Положив $u(a) = 0$ и $w_a = 0$ в оценке теоремы 6 мы получим новую оценку, которая учитывает лишь неполноценность $q(a)$ каждого алгоритма. Такая оценка будет менее точной, но в то же время она применима к произвольному множеству алгоритмов. Следовательно, необходимо сравнить два эффекта: увеличение вероятности переобучения при слабом замыкании множества A, и улучшение оценки при учете верхней связности. Так же представляет интерес сравнение оценки 6 и простой оценки, полученной с помощью неравенства Буля:

$$Q_\varepsilon(A) \leqslant \sum_{a \in A} H_L^{\ell, m}\left(\tfrac{\ell}{L}\,(m - \varepsilon k)\right),$$

где $m = n(a, \mathbb{X})$ — число ошибок алгоритма a на полной выборке.

Для проведения эксперимента было выбрано 8 задач из репозитория UCI. К каждой из восьми задач применялось четыре метода поиска логических закономерностей, реализованных в библиотеке Forecsys-LogicPro: случайный поиск (RND), случайный поиск с адаптацией весов признаков (RSA), генетический алгоритм (sGA) и поиск правил путём отбора признаков алгоритмом $TEMP$. Каждый метод поиска в процессе своей работы генерирует большое количество логических правил и вычисляет для них информативность. Следует подчеркнуть, что взаимодействие метода поиска с обучающей выборкой происходит только в момент вычисления информативности. Поэтому представляется разумным исследовать вероятность переобучения лишь для *наблюдаемой части семейства* — то есть для множества логических закономерностей, сгенерированных методом поиска в процессе своей работы. Все прочие логические закономерности предлагается объявить ненаблюдаемыми, и исключить из рассмотрения.

Ошибкой логических закономерностей считалось как покрытие объекта противоположного класса, так и непокрытие объекта своего класса. На рис. 2 приведены профили расслоения полученных семейств. Очевидно, что при минимизации числа ошибок на обучении вероятность выбрать алгоритм в результате обучения быстро падает с ростом числа его ошибок.

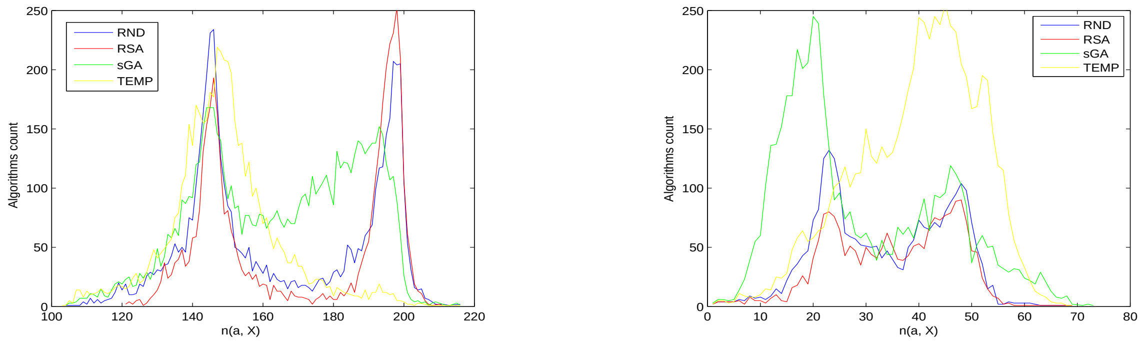

Рис. 2: Распределение алгоритмов по числу ошибок, задача Liver Disorders (слева) и Echo Cardiogram (справа).

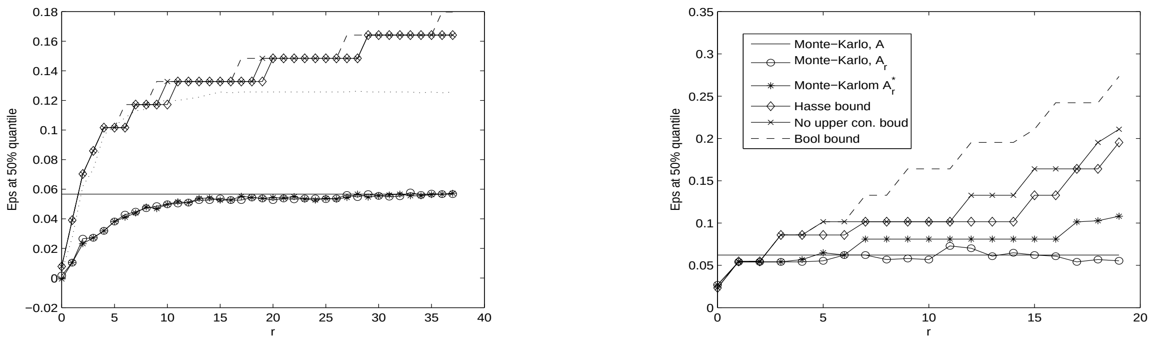

Рис. 3: Сравнение 0.5-квантили различных оценок вероятности переобучения, задача Liver Disorders (слева) и Echo Cardiogram (справа). Пунктирная кривая на левом рисунке соответствует семейству, полученному случайным перемешиванием векторов ошибок в A_r^*.

Поэтому в данном эксперименте рассматривались подмножества $A_r = \{a \in A\colon n(a,\mathbb{X}) \leqslant m_0+r\}$, где $m_0 = \min\limits_{a\in A} n(a,\mathbb{X})$. Значения r перебирались от 0 до 50. Для каждого A_r строилось внутреннее замыкание A_r^* и вычислялась 0.5-квантиль распределения вероятности переобучения, построенного методом Монте-Карло (по 10000 случайным разбиениям выборки на обучения и контроль). Также вычислялась 0.5-квантиль распределения, полученного по формулам расслоения связности — с учетом и без учета верхней связности.

Результаты сравнения приведены на рис. 3. Во-первых, видно что внутреннее замыкание множества алгоритмов может вести себя принципиально по-разному. Так, для задачи Liver Disorder вероятности переобучения A_r и A_r^* практически совпадают при всех значениях r. Для задачи Echo Cardiogram вероятность переобучения A_r^* оказывается заметно выше. Оценки вероятности переобучения также ведут себя по-разному. Для задачи Liver Disorder учёт связности не даёт улучшения по сравнению с оценкой Буля. Однако на задаче Echo Cardiogram улучшение становится очевидным.

Завышенность оценки 6 в задаче Liver Disorders можно объяснить плохим учетом связности (при малых r) и расслоения (при больших r). На левом рисунке пунктиром изображена кривая, соответствующая семейству с A_r^* со случайно переставленными ошибками. Данная процедура разрушает связность между алгоритмами. Видно, что при малых значениях r данная кривая хорошо приближает оценки теоремы 6. Плохой учет связности возникает из-за того, что в оценке рассматриваются лишь пары алгоритмов с вложенными векторами ошибок. Это, в частности, не позволяет учитывать связи между алгоритмами с равным числом ошибок. В то же время, теоретически доказано что эффект связности имеет место и в этом случае. В работах Ильи Толстихина и Александра Фрея была показана принципиальная разница между поведением вероятности переобучения для двух модельных семейств: максимально-компактного (центральное сечение шара), и максимально-разреженного (случайные подмножества слоя).

Вместе с тем, при больших значениях r оценка для перемешенного A_r^* стремится к горизонтальной асимптоте, в то время как оценка расслоения-связности продолжает расти. Это можно объяснить недостаточным учетом эффекта расслоения. В работах Евгения Соколова показано, что для произвольной пары алгоритмов a, b с различным числом ошибок алгоритм a с меньшим числом всегда уменьшает вероятность алгоритма b. Данный эффект наблюдается даже в тех случаях, когда вектор ошибок a не вложен в вектор ошибок b.

[ToDo] Причесать текст и добавить замечание о высокой толерантности обращения оценки к ошибке по сравнению с непосредственной оценкой среднего числа ошибок на контроле.

10 Заключение

[ToDo] Добавить заключение. Краткие выводы:

- Предложен способ учёта верхней связности, основанный на слабом замыкании множества алгоритмов;
- Показано, что учёт верхней связности может оказывать существенное влияние на качество оценки;
- Показано, что в ряде случаев оценки всё еще остаются завышенными. Проанализированы возможные причины завышенности и предложены дальнейшей способы борьбы с ними.